AMENDMENTS TO THE CLAIMS

1-83. (Cancelled)

84. (New) A method of facilitating executing an interpretive language in a system including a processing component, a memory component, and a hardware component, wherein the hardware component provides an interface between the processing component and the memory component, the method of the hardware component comprising:

receiving an address from the processing component;

determining whether the received address matches a fixed instruction fetch address;

determining whether the received address matches one of a plurality of fixed operand fetch addresses;

incrementing an interpreter language program counter and sending a current interpretive language address to the memory component when the received address matches the fixed instruction fetch address;

sending an operand address to the memory component when the received address matches one of the plurality of fixed operand fetch addresses;

sending the received address to the memory component when the received address fails to match the fixed instruction fetch address and any of the plurality of fixed operand fetch addresses; and

receiving data from the memory component based on one of the fixed instruction fetch address, the operand address, and the received address.

85. (New) The method of Claim 84, further including:

calculating an instruction jump address using the data and sending the instruction jump address to the processing component when the data is based on the fixed instruction fetch address.

86. (New) The method of Claim 84, further including:

storing the data and sending the data to the processing component using a predetermined bit order when the data is based on the operand address.

87. (New) The method of Claim 84, further including:

sending the data to the processing component when the data is based on the received address.

88. (New) The method of Claim 84, further including:

counting a number of times the received address matches the fixed instruction fetch address; and

when the number reaches a predetermined threshold, sending a thread switch jump address to the processing component.

89. (New) A hardware component that facilitates executing an interpretive language in a system, the system including processing component and a memory component, wherein the hardware component provides an interface between the processing component and the memory component, the hardware component comprising:

a first multiplexer for receiving an address from the processing component and providing an output to the memory component;

an interpreter language program counter for providing inputs of the first multiplexer; and

a decoding component for:

receiving the address,

comparing the received address to stored addresses, the stored addresses including a fixed instruction fetch address and a plurality of fixed operand fetch addresses, and

controlling the output of the first multiplexer based on a result of the comparing.

90. (New) The hardware component of Claim 89, wherein the decoding component sets the first multiplexer to provide the received address as the output of the first multiplexer when the received address fails to match any stored address.

91. (New) The hardware component of Claim 89, wherein the decoding component sets the first multiplexer to provide a current interpretive language address stored in the interpreter language program counter as the output of the first multiplexer when the received address matches the fixed instruction fetch address.

92. (New) The hardware component of Claim 89, wherein the decoding component sets the first multiplexer to provide an operand address stored in the interpreter language program counter as the output of the first multiplexer when the received address matches one of the plurality of fixed operand fetch addresses.

93. (New) The hardware component of Claim 89, further comprising:

a second multiplexer for receiving data from the memory component and providing an output to the processing component;

an instruction jump address generator component for receiving the data and providing inputs to the second multiplexer; and

an operand storing component for receiving the data, storing any operands of the data, and providing inputs to the second multiplexer,

wherein the decoding component controls the second multiplexer based on the result of the comparing.

94. (New) The hardware component of Claim 93, wherein the decoding component sets the second multiplexer to provide the data as the output of the second multiplexer when the received address fails to match any stored address.

95. (New) The hardware component of Claim 93, wherein the decoding component sets the second multiplexer to provide an output of the instruction jump address generator component as the output of the second multiplexer when the received address matches the fixed instruction fetch address.

96. (New) The hardware component of Claim 93, wherein the decoding component sets the second multiplexer to provide an output of the operand storing component as the output of the second multiplexer when the received address matches one of the plurality of fixed operand fetch addresses.

97. (New) The hardware component of Claim 93, further including a counter component for receiving an input from the decoding component and providing outputs to the second multiplexer and the decoding component.

98. (New) The hardware component of Claim 97, wherein the counter component counts a number of times the received address matches the fixed instruction fetch address, as determined by the decoding component, and

wherein when the number reaches a predetermined threshold, the counter component provides a thread switch jump address to the second multiplexer and the decoding component sets the second multiplexer to provide the thread switch jump address as the output of the second multiplexer.